IT & Business College of AIU

Department: Computer Science

Project Title: Obstacle Detecting Shoe for Visually Impaired People (ODSVIP)

Group Name: TEAM 1

Team members:

Rim Younes Elsayed 238715073

Ainuru Abilmukanova 238715057

Nurjanat Sultanova 238715019

Amir Imashov 238715051

Adilkhan Maksatov 238715060

Instructor: Mr. Mirlan Nurbekov

Group: SCA-23B

Date of submission: 17.12.2025

Fall semester of 2025/2026

Table of Contents

List of figures

List of tables

1 ABSTRACT

The Obstacle Detecting Shoe for Visually Impaired People (ODSVIP) prototype is an innovation to help the blind detect obstacles without relying entirely on white canes. The ODSVIP can help the blind community by sending a warning through a buzzer beep when there is an obstacle detected. Nowadays, especially in Kyrgyzstan, the blind usually uses white canes in their day-to-day lives. The use of white canes mostly focuses on detecting obstacles. Therefore, using our ODSVIP can solve this problem to a certain extent. In the ODSVIP, when the ultrasonic sensors are activated, they can detect obstacles in front within the distance the user of the device sets. Whenever the sensors detect an obstacle, it sends a signal to the Arduino UNO board, then the board sends a signal to the buzzer to make a sound to warn the person wearing the shoe that there is an obstacle ahead of him. Our device was created as an addition to the white cane, to make mobility smoother and more comfortable for those who have sight problems. The device consists of the hardware, which is the main functioning part of it, as well as a shoe accompanying it. All the wires and components were placed in a special sole, making the shoe look visually plausible. In addition to the device, we created an app to make the use of the device more convenient. In future development, we plan on replacing the sensors with more accurate ones and using vibration motors instead of buzzers.

2 Introduction & Background

2.1 Introduction

A person who is blind or visually handicapped may use a “white cane”. *A white cane is a navigation or identification aid for people living with sight loss. The use of a cane helps a person who is blind or partially sighted explore their surroundings, avoid obstacles and use tactile information to find their way, such as where to cross the road safely. It’s also an international symbol that lets others know that the user has a sight loss.* The market offers a wide range of canes, with the most popular choices for visually impaired people being Identification cane, Support cane and Mobility canes.

Some people who have lost their sight partially or completely feel comfortable walking without canes; however, they still use Identification Canes mainly to identify if there is an insignificant obstacle ahead of them or a staircase. Some canes, but not all of them, can be adjusted to the user’s preferences and height.

Although using canes is a common practice for the blind, its uses are limited. A cane is used mainly to examine the environment around the person using it, for any obstacles. The user may not accurately detect any stairs awaiting him by using only the cane. By research that had been done we came up with an idea of designing a shoe that may help people who have problems with their sight climb stairs comfortably. The innovation has no height limitations, which makes it fit for anyone. The device was created using 1 ultrasonic sesor, 1 buzzer, a battery, an Arduino UNO board, and lastly, a shoe model.

From what our research showed, there are no widely known smart shoes or devices on the market available for sale for the visually impaired. However, there is SmartShoe by Smart technology. The product uses a smart system for navigation assistance, and instead of a shoe, they have developed insoles for ordinary shoes.

2.2 Background

According to research done by the National Institutes of Medicine, visually impaired people have difficulties orienting when there are stairs. *VI was associated with an increased likelihood of having difficulties in managing stairs, especially in those who lived alone. This community-based study of people aged 60 years or older indicated that visually impaired people had greater difficulties in managing stairs compared with those without VI.*

Our project’s main goal is to aid the blind to walk about freely without any restrictions, and to help them manage stairs smoothly. This is because the current Smart Shoe market does not have much to offer for those with eyesight problems. For instance, they cannot warn the user of the stairs ahead of them, let alone help them to manage these kinds of obstacles.

3 Problem Statement & Objectives

3.1 Problem Statement

1. No specific device dedicated to helping visually impaired (VI) people detect obstructions in front of them without fully relying on white canes.
2. No device that would help people with VI manage stairs safely.
3. Visually impaired people always needing canes to let those around them that they have eyesight problems

3.2 Objectives

Objective of our project are to:

1. Aid the blind and VI people navigate through their ways safely without entirely needing canes and help them identifying stairs and managing them in a secure way for them.
2. Design a system that will make daily mobility for those with eyesight issues more convenient, by identifying indoor and outdoor obstacles/stairs.
3. Test the durability and effectiveness of our innovation to the VI community

3.3 Scope & Limitations

The ODSVIP was developed to be used as a tool to help those with VI navigate and manage stairs safely.

The scope in designing and developing ODSVIP:

1. Allow the blind and VI people to detect obstructions and stairs to help them
2. Obstruction detection within a limited range only, that can be set by the user using our app
3. The innovation is a rechargeable device

4 System requirements & constraints

4.1 Functional Requirements

1. The system must detect obstacles within a set range and alert the user of them
2. The system must warn the user about obstructions through a buzzer sound
3. The Arduino Uno board must receive signals from the ultrasonic sensor to trigger responses accordingly
4. The shoe model must house all the components in a way that will not affect the user's comfort
5. The mobile app of the project must connect to the hardware
6. The app must allow the users of the device to login and set their obstacle detection range preferences
7. The mobile app must indicate whether the device is on or off

4.2 Non-Functional Requirements

1. The ultrasonic sensor should be able to detect obstructions within the set range by the user (from 2 cm upwards)
2. The buzzer should make a sound to alert the user however its sound must not be too harsh or startling for the user
3. The mobile apps interface must be suitable with big buttons for visually impaired people
4. The system must work accurately both indoors and outdoors

4.3 System Constraints:

1. The device was built using an Arduino Uno board which limits the speed, memory, and performance of the device. Therefore, the system design must be simple.
2. All the hardware components must fit in the sole of the shoe, which limits the component’s size and placement.
3. The device fully relies on batteries, which limits continuous usage of the device without recharging.
4. The ultrasonic sensor has a limited range of obstacle detection. Environmental factors may affect the device’s accuracy.
5. The buzzer used has a limited volume control. To alert the user, the buzzer must be loud enough.
6. The mobile application depends on wireless connection, so any interference might affect the connectivity with the device.

5 System design (hardware & software)

5.1 System design overview

The system consists of a wearable smart shoe with a mobile application connected to it. The mobile application connects to the hardware wirelessly and using the app the user can control the settings like setting their desired obstacle detection range. The hardware processes the sensor data and alerts the user of obstacles ahead of him.

5.2 System flow:

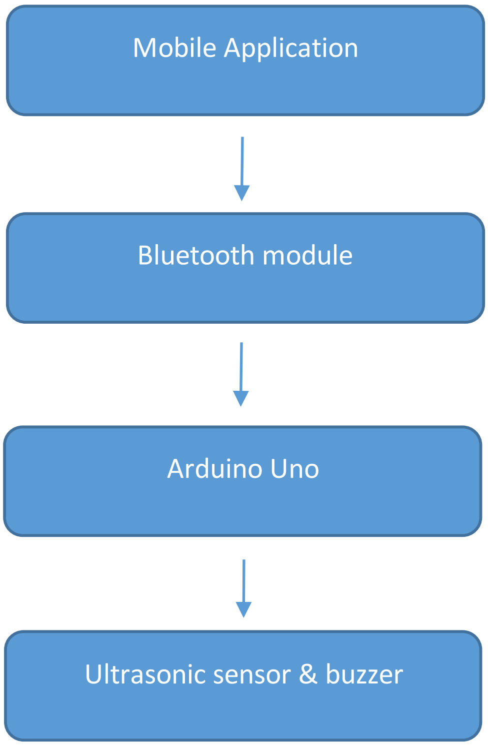


Figure 1.Sʜstem flow

The mobile application sends the user’s preferences to the Arduino board through the Bluetooth module. The Arduino board processes data from the sensor, and the buzzer makes sound when an obstacle is detected within the defined range.

5.3 Hardware Design

5.3.1 Main hardware components

1. Microcontroller: Arduino Uno
   Controls the entire system and processes data

Figure 2. Microcontroller Arduino Uno

2. Communication module: Bluetooth module HC-05
   Enables wireless communication between the mobile application and the hardware

Figure 3.Bluetooth module HC-05

3. Sensor: Ultrasonic sensor
   Measures the distance between the user and nearby obstacles

Figure 4.Ultrasonic sensor

4. Actuator: buzzer
   When an obstacle is detected, it produces an alert sound

Figure 5.Buzzer

5. Power supply: battery
   Provides power to the Arduino board and other components

Figure 6.Battery

5.3.2 Signal flow

The system's signal works as follows:

1. The ultrasonic sensor detect the distance to an obstacle
2. The sensor sends data to the Arduino board
3. The Arduino board compares the readings with the user-defined range preferences from the mobile application
4. If there is an obstacle within the set range the Arduino sends signal to the buzzer to alert the user
5. The mobile app displays the device status (on/off) and allows the user to update settings

5.3.3 Voltage levels and power

1. The Arduino Uno operates at 5V
2. The ultrasonic sensor and buzzer operate at 5V
3. The HC-05 module is powered using 5V
4. All components are powered using a battery integrated in the shoe design

5.4 Software Design

5.4.1 Software flow:

The state diagram below represents how the system works. Firstly, the system enters the initialization state, where all the components and Bluetooth communications are set up. The system then moves on to the Bluetooth connection state, where it receives the range of detection preferences data from the mobile application. After receiving the data, the system then moves to the measure data state, where the ultrasonic checks the area for any nearby obstacles. If the measured distance is within the range that the user sets in the app, the system transitions to the obstacle detected state and turns on the buzzer. If no obstacle is detected, the system moves to the no obstacle detected state and keeps the buzzer off. The system continuously loops back to the distance measurement state.

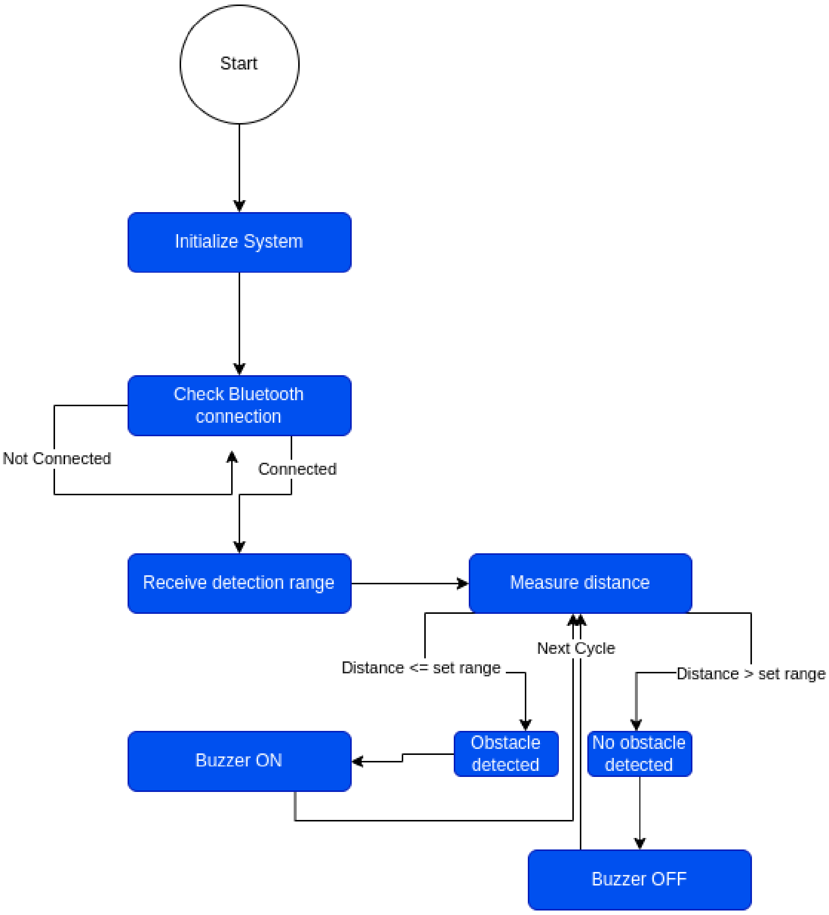


Figure 7. Software flow

5.4.2 Software function explanation

Setup and Initialization

1. Initialize Arduino pins
2. Setup bluetooth communication
3. Configure the ultrasonic sensor and the buzzer
4. Set default detection range

Reading inputs

1. Recieve user detection range preferences from the mobile application
2. Read distance data from the ultrasonic sensor

Desicion logic

1. Compare the distance with the user-defined range
2. Decide whether there is an obstacle or not

Writing outputs

1. Activate the buzzer if there is an obstacle
2. Keep the buzzer off if no obstacles are detected
3. Update the device status in the mobile application

6 Prototypes & progress

6.1 Prototype 0000

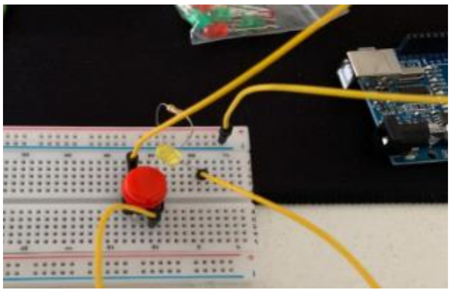


Figure 8. Prototype 0000 - LED and button connected to Arduino Uno.

Purpose

The prototype shows how it reacts when someone unsteady bumps into something. Hit the button – that's like a collision, yet the LED glows once the system notices. For the finished model, we'll swap the light for a buzzer, so the user hears the alert right away.

Description

The prototype runs on an Arduino Uno for control. One digital pin takes input from a push button, while another sends output to an LED using a resistor. Pressing the button lets the board sense it - then lights up the LED right away. That basic action shows how the finished version could spot barriers and respond instantly.

Table 1 - Prototype 0000 components

Component	Amount	Function
Arduino Uno	1	Main microcontroller
LED	1	Visual output indicator
Button	1	Simulated obstacle detection input
Resistor (220 Ω)	1	Protects the LED from high current
Breadboard	1	Assembly platform
Jumper wires	several	Electrical connection
USB power cable	1	Provides power to the Arduino Uno

How it works?

When someone hits the button, the Arduino picks up a HIGH signal at its input pin. Right after that, it turns on the LED - just so you can see something happened right away. That shows how basic steps work together: press → process → light, setting things up for adding sensor responses down the line.

6.2 Prototype 0001

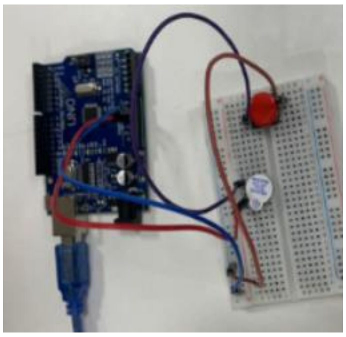


Figure 9.Prototype 0001 - buzzer and button connected to Arduino Uno.

Purpose

This prototype tests a tweaked feedback setup. Rather than using an LED, it uses a buzzer so you can hear when the shoe finds something in front. It's meant to see whether the sound kicks in right after hitting an object - which here means pressing a button. The main point is making sure that buzzer starts exactly when needed.

Description

In this version, the system is still based on an Arduino Uno and a push button. The button is used to simulate a collision with an obstacle. When the button is pressed, the Arduino activates the buzzer, producing a sound alert. This prototype demonstrates how audio feedback can be used instead of visual signals in the final device.

Table 2 - Prototype 0001 components

Component	Amount	Function
Arduino Uno	1	Main microcontroller
Buzzer	1	Audio feedback when an obstacle is detected---
Button	1	Simulated obstacle detection input
Breadboard	1	Assembly platform
Jumper wires	several	Electrical connection
USB power cable	1	Provides power to the Arduino Uno

How it works?

When the button is pressed, the Arduino receives a signal and immediately activates the buzzer. The sound indicates that an obstacle has been detected. This simple process shows how the system can warn the user using audio feedback instead of light.

6.3 Prototype 0002

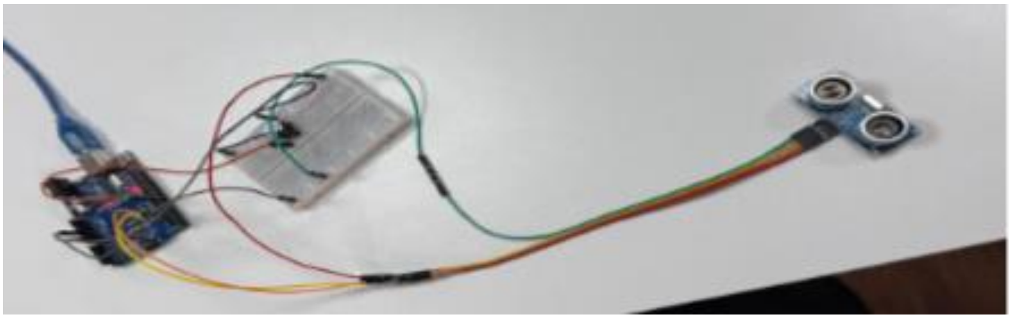


Figure 10. Prototype 0002 - Ultrasonic Sensor for Obstacle Detection

Purpose

This prototype was created to replace the temporary button with a real distance-detecting sensor. The goal was to test whether the ultrasonic sensor can accurately detect obstacles and trigger feedback for the user, making the system much closer to the final shoe design.

Description

In this version of the prototype, the push button from earlier stages was replaced with an ultrasonic sensor. The sensor continuously measures the distance to objects in front of it. When something gets too close, the sensor sends a signal to the Arduino Uno. Depending on the distance, the Arduino activates the buzzer to alert the user. This prototype demonstrates actual obstacle detection rather than simulated impact.

Table 3 - Prototype 0002 components

Component	Amount	Function
Arduino Uno	1	Main microcontroller
Buzzer	1	Audio feedback when an obstacle is detected---
Ultrasonic sensor (HC-SR04)	1	Detects distance to obstacles
Breadboard	1	Assembly platform
Jumper wires	several	Electrical connection
USB power cable	1	Provides power to the Arduino Uno

How it works?

The ultrasonic sensor sends sound waves and measures how long they take to return after hitting an object. If the measured distance becomes too small (meaning an obstacle is close), the Arduino processes this information and activates the buzzer as an alert. This prototype replaces the simulated “button hit” with real, automated distance detection.

6.4 Prototype 0003

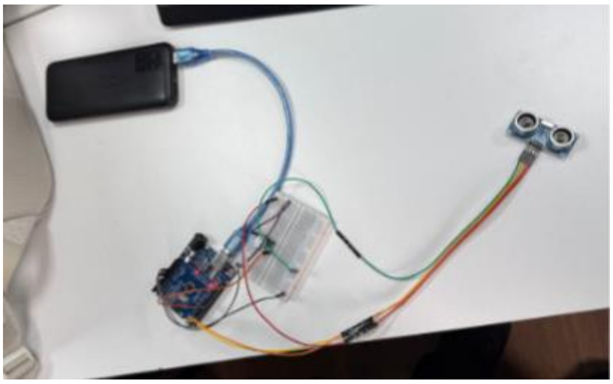


Figure 11.Prototype 0003 - the device operating using a portable power source.

.

Purpose

This prototype was created to make the system fully portable. By adding a power bank as the power source, the device can now be used while walking, without needing to stay connected to a laptop or external power supply. This brings the prototype much closer to a real wearable obstacle-detecting shoe.

Description

In this version, a power bank was added to power the Arduino Uno and all connected components. The USB cable from the power bank supplies constant 5V, allowing the ultrasonic sensor and the buzzer to work anywhere. This upgrade tests how stable the system is during movement and whether the sensor continues working correctly while the user is walking.

Table 4 - Prototype 0003 components

Component	Amount	Function
Arduino Uno	1	Main microcontroller
Buzzer	1	Audio feedback when an obstacle is detected
Ultrasonic sensor (HC-SR04)	1	Detects distance to obstacles
Breadboard	1	Assembly platform
Jumper wires	several	Electrical connection
USB power cable	1	Provides power to the Arduino Uno

How it works?

The power bank provides continuous 5V to the Arduino through a USB cable. Once powered, the ultrasonic sensor starts measuring distance automatically. When an obstacle is close enough, the Arduino activates the buzzer to warn the user. Because the system no longer depends on a computer, it can now be tested in real walking conditions.

6.5 Prototype 0004

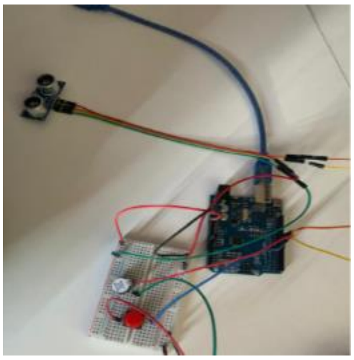


Figure 12.Prototype 0004 - system with an on/off power switch.

Purpose

This prototype was created to improve power efficiency and battery life. By adding an on/off switch, the user can manually control when the system is active, preventing unnecessary battery drain when the device is not in use.

Description

In this version of the prototype, a physical on/off switch was added between the power bank and the Arduino Uno. The switch allows the user to turn the system on before use and completely power it off afterward. This makes the device more practical for everyday use and better suited for a wearable obstacle-detecting shoe.

Table 5 - Prototype 0004 components

Component	Amount	Function
Arduino Uno	1	Main microcontroller
Buzzer	1	Audio feedback when an obstacle is detected
Ultrasonic sensor (HC-SR04)	1	Detects distance to obstacles
Breadboard	1	Assembly platform
Jumper wires	several	Electrical connection
USB power cable	1	Provides power to the Arduino Uno
On/Off switch button	1	Controls system power

How it works?

When the switch is turned on, the power bank supplies electricity to the Arduino and the system starts operating normally. The ultrasonic sensor begins detecting obstacles, and feedback is provided through the buzzer or vibration motor. When the switch is turned off, power to the system is completely cut, helping to conserve battery life.

6.6 Prototype 0005

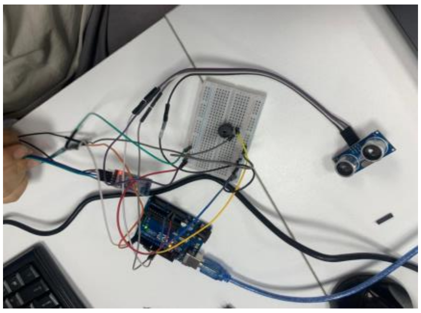


Figure 13.Prototype 0005 – system with HC-05 Bluetooth module for mobile control

Purpose

This prototype allows the device to be controlled wirelessly through a mobile application using the HC-05 Bluetooth module. Users can change the obstacle detection distance, turn the system on or off, and manage user and admin access directly from the app.

Description

In this version, the HC-05 Bluetooth module is connected to the Arduino Uno, enabling communication with a mobile application. The ultrasonic sensor continuously measures the distance to obstacles. Through the app, the user can adjust the detection distance (default 50 cm), so the buzzer activates only when an obstacle is within the set range. The application also allows user and admin registration and lets the user turn the system on or off remotely.

Table 6 - Prototype 0005 components

Component	Amount	Function
Arduino Uno	1	Main microcontroller
Buzzer	1	Audio feedback when an obstacle is detected
Ultrasonic sensor (HC-SR04)	1	Detects distance to obstacles
Breadboard	1	Assembly platform
Jumper wires	several	Electrical connection
USB power cable	1	Provides power to the Arduino Uno
HC-05 Bluetooth module	1	Enables wireless communication with the mobile app

How it works?

The Arduino reads distance data from the ultrasonic sensor continuously. The HC-05 module communicates with the mobile application, allowing the user to change the detection distance or turn the system on and off. If an obstacle comes closer than the set distance, the buzzer or vibration motor activates, giving immediate feedback to the user.

Mobile application functions

- Change obstacle detection distance
- Turn the device on and off
- User and admin registration

- Manage user access and settings

6.7 Prototype 0006 - Final prototype

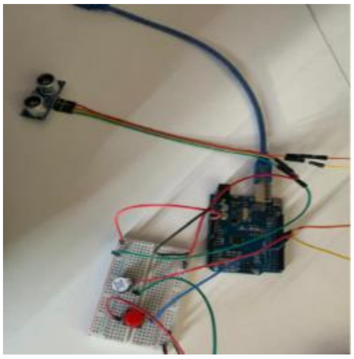


Figure 14.. Prototype 0006 - final obstacle-detecting shoe with integrated hardware

Purpose

This prototype represents the final version of the project, where all system components are fully integrated into a shoe. The goal is to demonstrate how the device can be used in real-life conditions while walking.

Description

In this final prototype, all hardware components are placed inside a shoe model. The shoe body is made from a box material, and the sole is intentionally thick to accommodate the electronic components. The Arduino, power bank, and other hardware are hidden inside the sole, while the ultrasonic sensor is positioned at the front of the shoe through a small opening. This layout protects the electronics and allows the sensor to detect obstacles ahead of the user.

Table 7 - Prototype 0006 components

Component	Amount	Function
Arduino Uno	1	Main microcontroller
Buzzer	1	Audio feedback when an obstacle is detected
Ultrasonic sensor (HC-SR04)	1	Detects distance to obstacles
Breadboard	1	Assembly platform
Jumper wires	several	Electrical connection
USB power cable	1	Provides power to the Arduino Uno
Shoe model	1	Holds and protects all hardware
HC-05 Bluetooth module	1	Enables wireless communication with the mobile app

How it works?

When the system is turned on, the ultrasonic sensor measures the distance to obstacles in front of the shoe. If an object is detected within the set distance, the system alerts the user through sound or vibration. The user can adjust settings, such as detection distance, through the mobile application.

7 Implementation (code + building the project)

7.1 Overview of the implementation

This section describes how the ODSVIP obstacle-detecting shoe was actually built and tested in practice. It outlines the main steps taken to assemble the hardware, write the Arduino program and connect the system to a mobile application using bluetooth. The emphasis here is on the real development process rather than theoretical concepts.

7.2 Hardware implementation

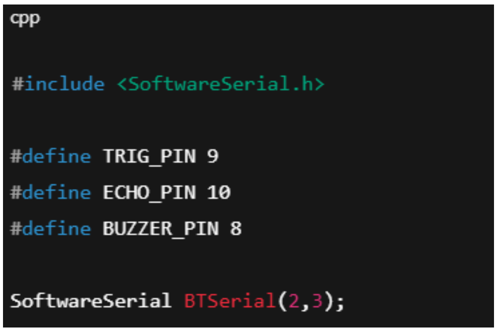
The Arduino Uno was selected as the main controller because it is stable, accessible and easy to program. HC-SR04 ultrasonic sensor was placed at the front of the shoe so it could detect obstacles in the user's walking path. Buzzer was connected to the Arduino to provide an audible warning whenever an object is detected within the defined distance.

For wireless interaction, HC-05 bluetooth module was connected to the Arduino through serial pins. The device is powered by a portable power bank, which allows it to work without being connected to an external power source. On/off switch was added between the power bank and the Arduino, giving the user control over power usage and helping to extend battery life.

All electronic parts were fitted inside the shoe structure. Sole was made thicker than normal to safely hold the components, and a small opening was created at the front so the ultrasonic sensor could function correctly. This design keeps the hardware protected and makes our shoe suitable for regular use.

7.3 Software implementation - Arduino code

7.3.1 Library and pin definition

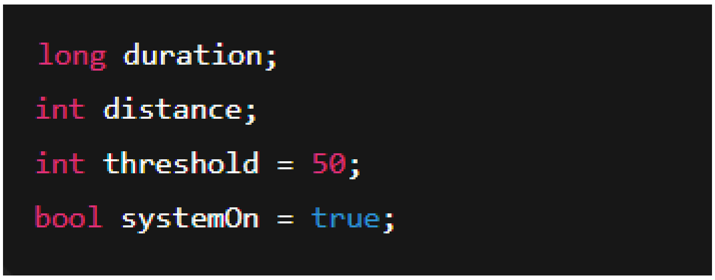


```cpp
#include <SoftwareSerial.h>

#define TRIG_PIN 9
#define ECHO_PIN 10
#define BUZZER_PIN 8

SoftwareSerial BTSerial(2,3);
```

Figure 15

This part defines the libraries and assigns pins for all components. The ultrasonic sensor, buzzer, and Bluetooth module are connected to specific pins on the Arduino Uno. A software serial connection is also created to allow communication with the HC-05 Bluetooth module. This setup ensures all hardware parts can communicate properly and the system can function as intended.

7.3.2 Global variables and system parameters

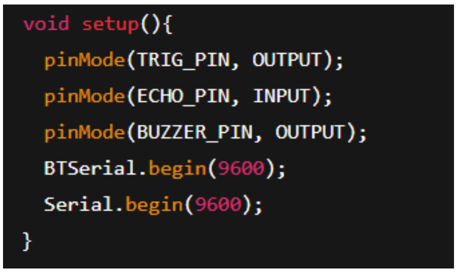


```
long duration;
int distance;
int threshold = 50;
bool systemOn = true;
```

Figure 16

Here, the variables used by the Arduino program are defined. These include the distance measured by the sensor, the threshold for obstacle detection, and a system state variable to control whether the device is on or off. Defining these variables at the start allows the program to run smoothly and respond correctly to sensor readings and user commands.

7.3.2 Setup function

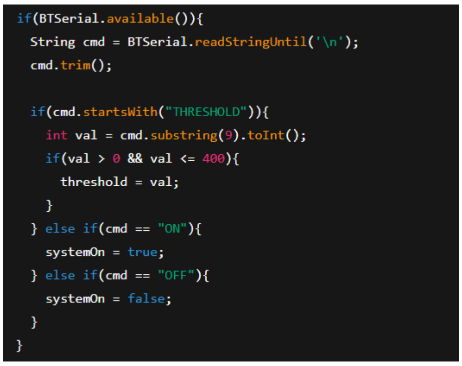


```
void setup(){
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  BTSerial.begin(9600);
  Serial.begin(9600);
}
```

Figure 17

The setup function initializes the Arduino pins and starts serial communication for both Bluetooth and debugging. This ensures the ultrasonic sensor and buzzer are ready to operate and the mobile application can connect to the HC-05 module without issues. Proper initialization is essential for reliable system performance.

7.3.3 Bluetooth command proccessing

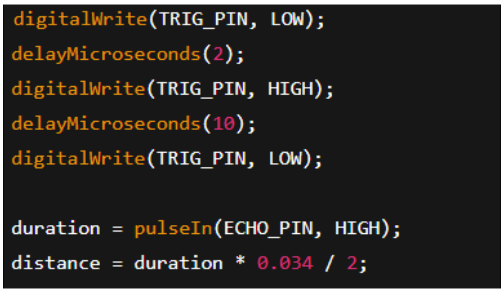


```
if(BTSerial.available()){
  String cmd = BTSerial.readStringUntil('\n');
  cmd.trim();

  if(cmd.startsWith("THRESHOLD")){
    int val = cmd.substring(9).toInt();
    if(val > 0 && val <= 400){
      threshold = val;
    }
  } else if(cmd == "ON"){
    systemOn = true;
  } else if(cmd == "OFF"){
    systemOn = false;
  }
}
```

Figure 18

This section handles incoming commands from the mobile application. Users can change the obstacle detection distance and turn the system on or off remotely. Processing these commands allows the device to be adjusted according to personal preferences and environmental conditions.

7.3.4 Ultrasonic distance measurement

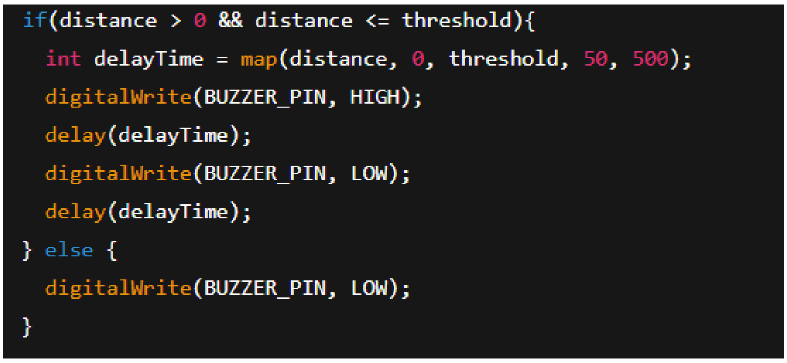


```
digitalWrite(TRIG_PIN, LOW);
delayMicroseconds(2);
digitalWrite(TRIG_PIN, HIGH);
delayMicroseconds(10);
digitalWrite(TRIG_PIN, LOW);

duration = pulseIn(ECHO_PIN, HIGH);
distance = duration * 0.034 / 2;
```

Figure 19

The ultrasonic sensor sends pulses and measures the time it takes for the echo to return. This time is then converted into a distance in centimeters. The measurement allows the system to know how far obstacles are from the user, forming the basis for activating alerts.

7.3.5 Obstacle detection and buzzer control

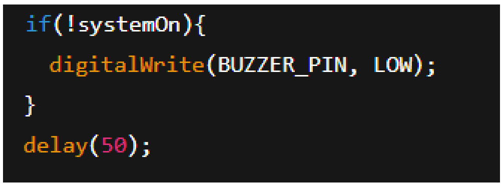


```
if(distance > 0 && distance <= threshold){
  int delayTime = map(distance, 0, threshold, 50, 500);
  digitalWrite(BUZZER_PIN, HIGH);
  delay(delayTime);
  digitalWrite(BUZZER_PIN, LOW);
  delay(delayTime);
} else {
  digitalWrite(BUZZER_PIN, LOW);
}
```

Figure 20

When an obstacle is detected within the set threshold distance, the buzzer is activated. The closer the obstacle, the faster the buzzer beeps. This feedback gives the user an intuitive sense of how near the obstacle is, helping them navigate safely.

7.3.6 System state handling



```
if(!systemOn){
  digitalWrite(BUZZER_PIN, LOW);
}
delay(50);
```

Figure 21

This part ensures that when the system is turned off, the buzzer and obstacle detection are disabled. It also includes a short delay to stabilize the system and prevent false triggers. Managing the system state properly ensures both safety and energy efficiency.

The program also receives commands from the bluetooth module. These commands allow the user to adjust the detection distance and turn system on or off through the mobile application. This adds flexibility and makes device easier to personalize. The full Arduino code is provided in the Appendix.

7.4 Mobile Application implementation

Mobile application was created to allow users to control the device remotely. The app includes a basic login system with separate user and admin roles for access management. After logging in, the application searches for the HC-05 Bluetooth module and connects to it automatically. Through the app, users can change the obstacle detection distance, switch the device on or off, and adjust simple settings. This makes it possible to adapt the shoe to different environments and individual needs, improving overall comfort and usability.

7.5 Communication between application and hardware

Communication between mobile application and the Arduino Uno is handled through bluetooth serial communication. Application sends commands such as updated distance values or power control instructions to the Arduino. Arduino then processes these commands and updates the system's behavior accordingly. This two-way communication allows the system to respond quickly and reliably.

7.6 Building and integration

After completing hardware and software development separately, all components were integrated into a single working system and tested together. The Arduino, sensors, buzzer, bluetooth module, and power supply were securely fixed inside the shoe to prevent movement during walking. The final prototype demonstrates a functional wearable device that can detect obstacles and provide real-time feedback, supporting safer movement for visually impaired users.

8 Testing & Results

8.1 Testing overview

Testing was conducted to ensure that the device meets the functional and non-functional requirements for the system. All the hardware and software components were tested individually to verify that they were working, and, lastly, they were tested as part of a complete system.

8.2 Test cases & Results

1. Ultrasonic sensor testing

Table 8 - Ultrasonic test

Objective	Method	Expected Results	Actual Results	Status
Verify distance measurement	Place object at different distances	Object detected	Object detected	passed
Minimum detection range	Place object at 2 cm distance	Object detected	Object detected	passed
Indoor environment testing	Test sensor indoors	Stable readings	Stable readings	passed
Outdoor environment testing	Test sensor outdoors	Stable readings	Slight inaccuracy	passed

2. Buzzer testing

Table 9 - Buzzer test

Objective	Method	Expected Results	Actual Results	Status
Verify buzzer activation	Trigger obstacle within range	Buzzer ON	Buzzer activated	passed
Verify buzzer deactivation	Remove obstacle	Buzzer OFF	Buzzer deactivated	passed
Test sound comfort level	Observe buzzer sound	Sound is audible but not too harsh	Sound acceptable	passed

3. Bluetooth module testing

Table 10 - Bluetooth module test

Objective	Method	Expected Results	Actual Results	Status
Bluetooth connection	Connect mobile app to HC-05	Successful connection	Connected successfully	passed
Data transmission	Send range value from app	Arduino receives data	Data received correctly	passed
Connection stability	Maintain connection during use	Stable connection	Minor interference outdoors	passed

4. Mobile application

Table 11 - Mobile app test

Objective	Method	Expected Results	Actual Results	Status
Verify login functionality	Log in using the mobile app	Successful login	Login successful	passed
Test obstacle range setting	Change detection range in the app	Range updated on device	Range updated correctly	passed
Verify device status display	Check ON/OFF indicator	Correct status shown	Status displayed correctly	passed

5. Overall system testing

Table 12 - Overall system test

Objective	Method	Expected Results	Actual Results	Status
End-to-end system functionality	Test full system operation	Correct obstacle alerts	System worked as expected	passed
Indoor system testing	Use device indoors	Acceptable performance	Reliable detection observed	passed
Outdoor system testing	Use device outdoors	Acceptable performance	Minor accuracy variation observed	passed

8.3 Summary of results

By testing it was confirmed that all the components functioned properly both individually and as part of a complete system. Minor limitations were observed during outdoor testing which can be addressed in future improvements.

9 Discussion

9.1 What worked well

The ODSVIP performed well during testing. The ultrasonic sensor was able to detect obstructions within the user-defined range and send signals to the buzzer to alert the user. The Bluetooth connection with the application was successful. The user-defined preferences were able to be transmitted to the Arduino board. The buzzer was able to alert the user about obstacles without sounding too harsh. The mobile application interface was easy to use, especially for visually impaired users due to its large buttons.

9.2 Problems faced during development

When developing the device, we faced a few problems. The ultrasonic sensors' readings were inaccurate due to noise during outdoor testing; thus, it gave the buzzer a false signal to alert the user of obstacles. Power management was a concern as well, since we had to use a small battery so as not to interfere with how the show looked. The Bluetooth connection sometimes was not stable due to interference.

9.3 Prototype evolution

The first prototype consisted of a button and an LED, and that is when the idea of creating ODSVIP was born. The button served as a collusion detector and the LED as an alerting system. As our project evolved, we replaced the button with an ultrasonic sensor and the LED with a buzzer. In the final versions of the device, we decided to create an application to make the use of the device even more convenient. Using the app the person using the device can set his own detection range preferences.

9.4 Trade-offs

During the design process of the device several trade-offs were made. Due to limited budget we had to use an ultrasonic sensor instead of a more advanced sensor, which affected the accuracy of obstacle detection in some environments. Battery size was limited, which resulted in a shorter usage time. The buzzer was chosen over a vibration motor / audio guidance, even though this would be more suitable for visually impaired people. These trade-offs allowed the device to maintain simplicity, and it kept being affordable in this way.

10 Conclusion & Future work

10.1 Conclusion

Safety for visually impaired communities should be emphasized and brought up in our daily life. More facilities or special help should be prepared for them to ensure their safety in daily life. To conclude things, our project addressed the issue of providing an obstacle detecting smart shoe for the visually impaired and blind. The final system was able to detect obstacles within the user-defined range and send an alert to the user through a buzzer sound whenever anything was detected. The system successfully integrates an ultrasonic sensor, Arduino Uno, HC-05 Bluetooth module, and a mobile application. Users can set their desired obstacle detection range and check whether the device is on or off through the mobile app. Testing results showed that the system works well in both indoor and outdoor environments, fulfilling the main functional and non-functional requirements.

10.2 Future work

Although the system works as intended, several improvements could be made. Instead of detecting obstacles only ahead of the user, we want to add another sensor to the bottom of the shoe so it can detect holes/ cliffhangers or stairs. For that, we would need to add one more sensor. After many observations, we concluded that an ultrasonic sensor was not the best fit for this type of device, so we want to replace it with a more accurate sensor. Since the sound of buzzers might get irritating over time, we want to replace the buzzer with vibration motors. Regarding the mobile application, we want to integrate a sound navigating system for those who can’t see completely. A more compact and durable enclosure can be designed to improve comfort and protect the hardware. Cloud connectivity could also improve the device even more, by providing usage data storage and real-time monitoring.

11 REFERENCES

Mahanty, S., Karunan, M., Sayyad, I. and Khursade, S. (2021) *Smart shoes for visually impaired*. International Journal of Creative Research Thoughts, 9(2), February. Available at: https://www.ijcrt.org/papers/IJCRT2102049.pdf

Khanam, A., Dubey, A. and Mishra, B. (2018) *Smart shoes for visually impaired people*. International Journal of Advanced Research in Science and Engineering, 7(Special Issue 01), April. Available at: http://www.ijarse.com/images/fullpdf/1522382800_BIT735ijarse.pdf

Wikimedia Foundation (2022) *White cane*. Wikipedia. Available at: https://en.wikipedia.org/wiki/White_cane (Accessed: 9 December 2025).

World Health Organization (n.d.) *Vision impairment and blindness*. World Health Organization. Available at: https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment (Accessed: 9 December 2025).

Michael, K.M., Gresset, J., Jones, L.A. and Witt, J.P. (2015) 'Stair-climbing difficulties in older adults with vision impairment', *Clinical Interventions in Aging*, 10, pp. 1787–1794. Available at: https://pmc.ncbi.nlm.nih.gov/articles/PMC4636376/

CNIB (n.d.) *The white cane*. Available at: https://www.cnib.ca/en/about-white-cane (Accessed: 15 December 2025).

12 Appendix A

Full Arduino Uno code:

Prototype 0000 - Button with LED

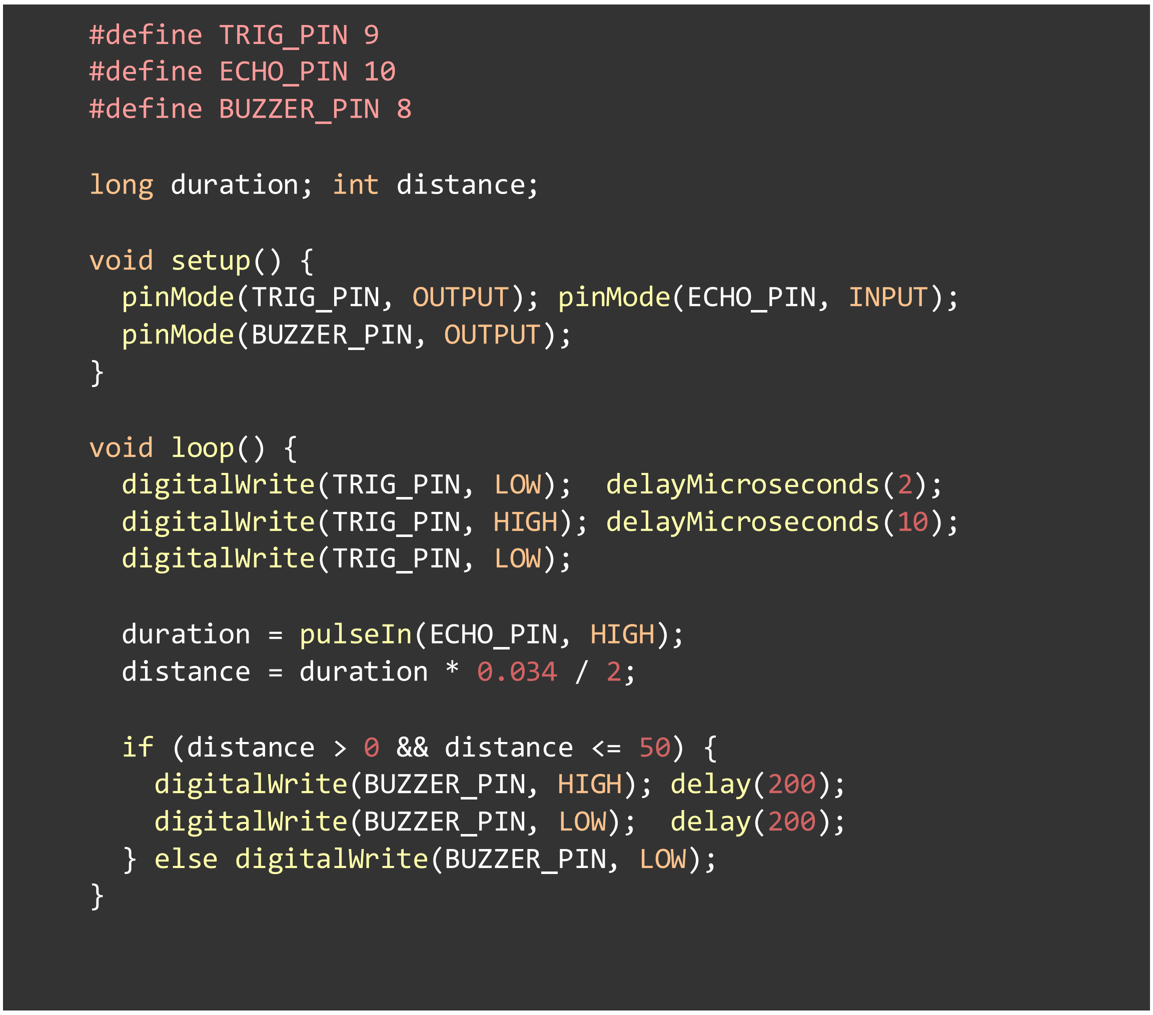
This prototype does not include software code. The LED was activated through a simple electrical circuit to demonstrate basic system feedback.

Prototype 0001 - Button with Buzzer

This prototype also does not use Arduino code. The buzzer was triggered directly through the circuit to test audio feedback.

Prototype 0002 - Ultrasonic Sensor + Buzzer

This prototype uses Arduino code to measure distance and activate the buzzer when an obstacle is detected.



```
#define TRIG_PIN 9
#define ECHO_PIN 10
#define BUZZER_PIN 8

long duration; int distance;

void setup() {
  pinMode(TRIG_PIN, OUTPUT); pinMode(ECHO_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
}

void loop() {
  digitalWrite(TRIG_PIN, LOW);  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH); delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  duration = pulseIn(ECHO_PIN, HIGH);
  distance = duration * 0.034 / 2;

  if (distance > 0 && distance <= 50) {
    digitalWrite(BUZZER_PIN, HIGH); delay(200);
    digitalWrite(BUZZER_PIN, LOW);  delay(200);
  } else digitalWrite(BUZZER_PIN, LOW);
}
```

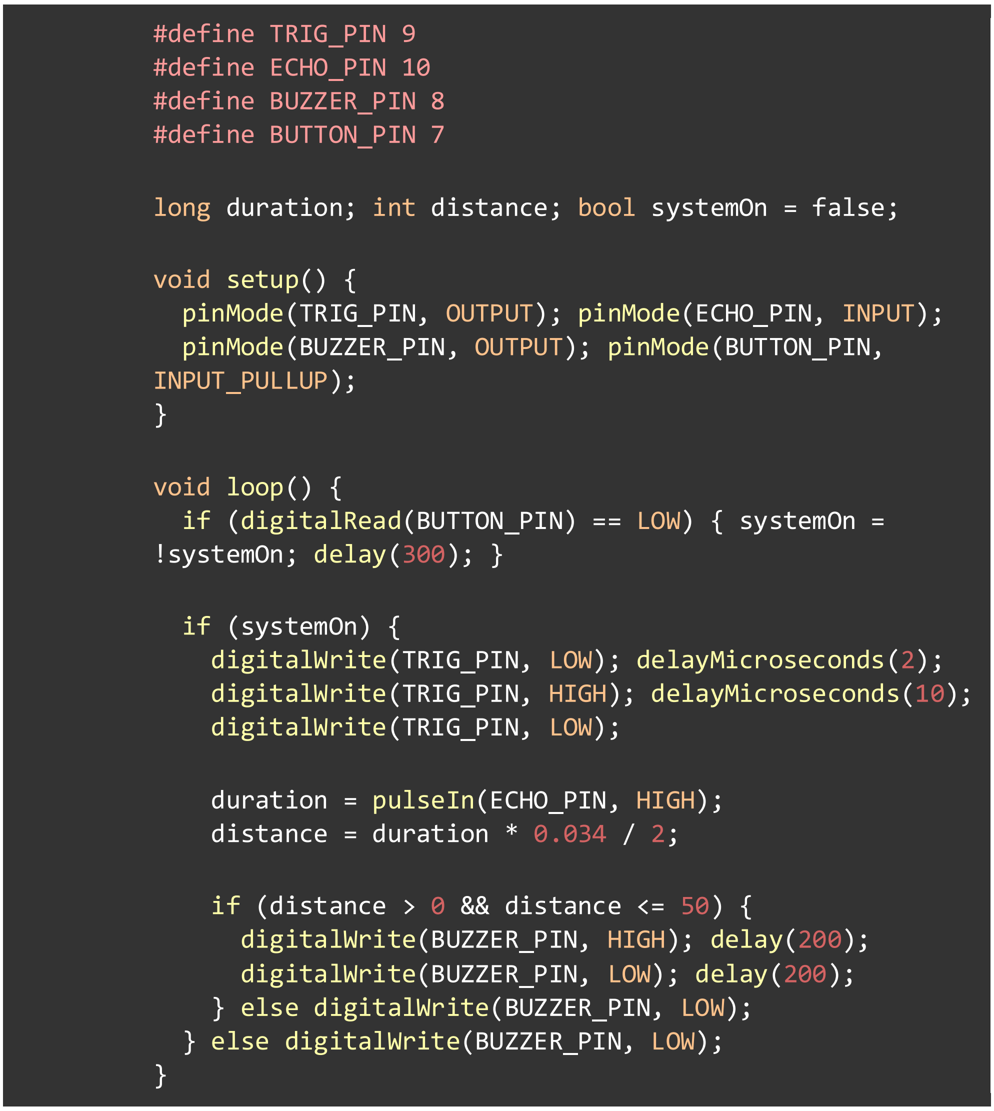
Figure 22

Prototype 0003 - Portable Power Supply

The same Arduino code as in Prototype 0002 is used. The difference is the power source, which was changed to a power bank.

Prototype 0004 - On/Off Switch

This version includes minor code adjustments to support power control.



```
#define TRIG_PIN 9
#define ECHO_PIN 10
#define BUZZER_PIN 8
#define BUTTON_PIN 7

long duration; int distance; bool systemOn = false;

void setup() {
  pinMode(TRIG_PIN, OUTPUT); pinMode(ECHO_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT); pinMode(BUTTON_PIN,
INPUT_PULLUP);
}

void loop() {
  if (digitalRead(BUTTON_PIN) == LOW) { systemOn =
!systemOn; delay(300); }

  if (systemOn) {
    digitalWrite(TRIG_PIN, LOW); delayMicroseconds(2);
    digitalWrite(TRIG_PIN, HIGH); delayMicroseconds(10);
    digitalWrite(TRIG_PIN, LOW);

    duration = pulseIn(ECHO_PIN, HIGH);
    distance = duration * 0.034 / 2;

    if (distance > 0 && distance <= 50) {
      digitalWrite(BUZZER_PIN, HIGH); delay(200);
      digitalWrite(BUZZER_PIN, LOW); delay(200);
    } else digitalWrite(BUZZER_PIN, LOW);
  } else digitalWrite(BUZZER_PIN, LOW);
}
```

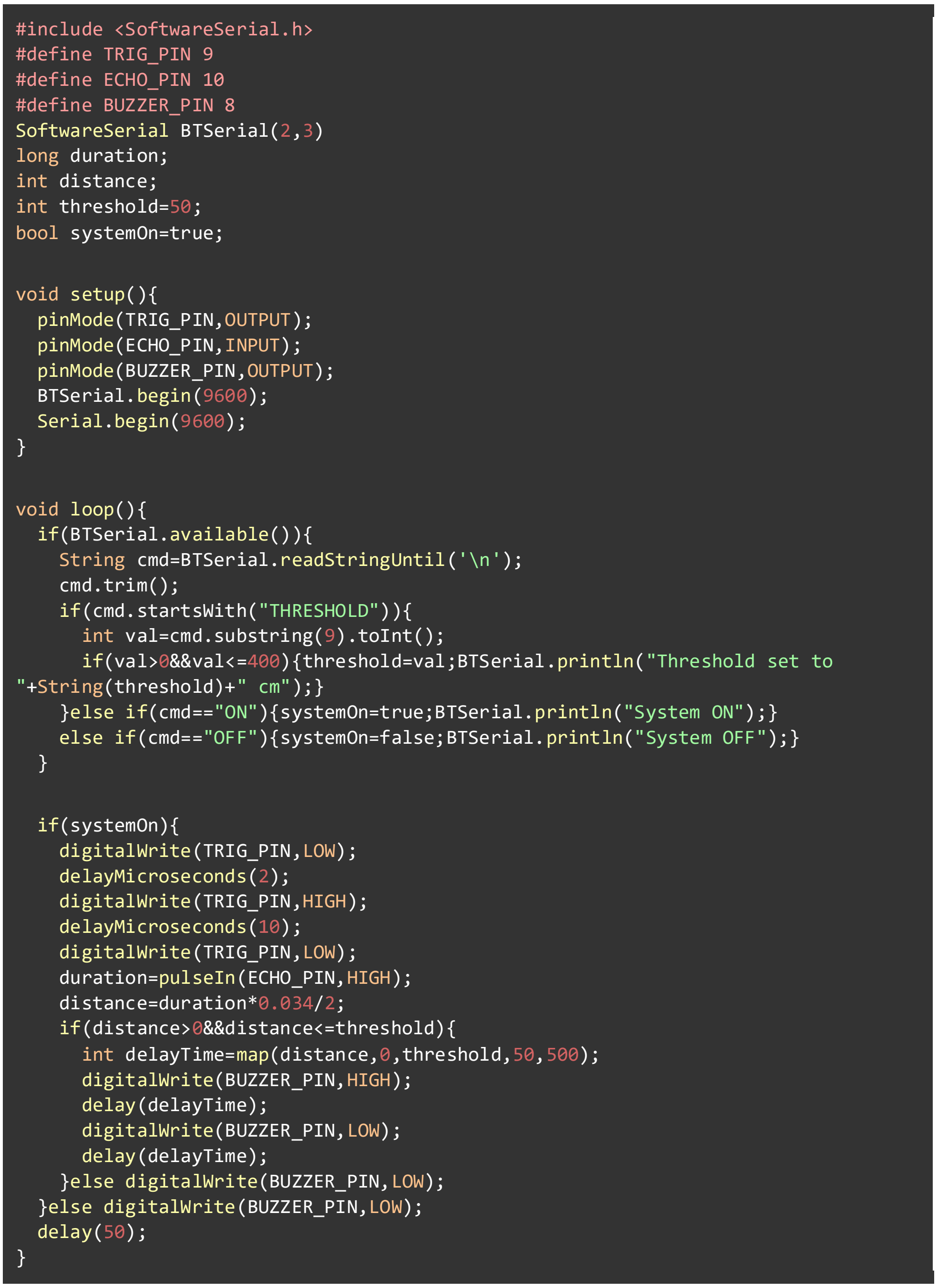
Figure 23

Prototype 0004 - Bluetooth Control via Mobile Application (HC-05)

This prototype adds the HC-05 Bluetooth module to enable wireless control through a mobile application. Users can adjust obstacle detection distance, turn the device on or off, and manage user and admin accounts directly from the app.

Mobile Application Functions:

- Change obstacle detection distance
- Turn the device on and off
- User and admin registration
- Manage user access and settings

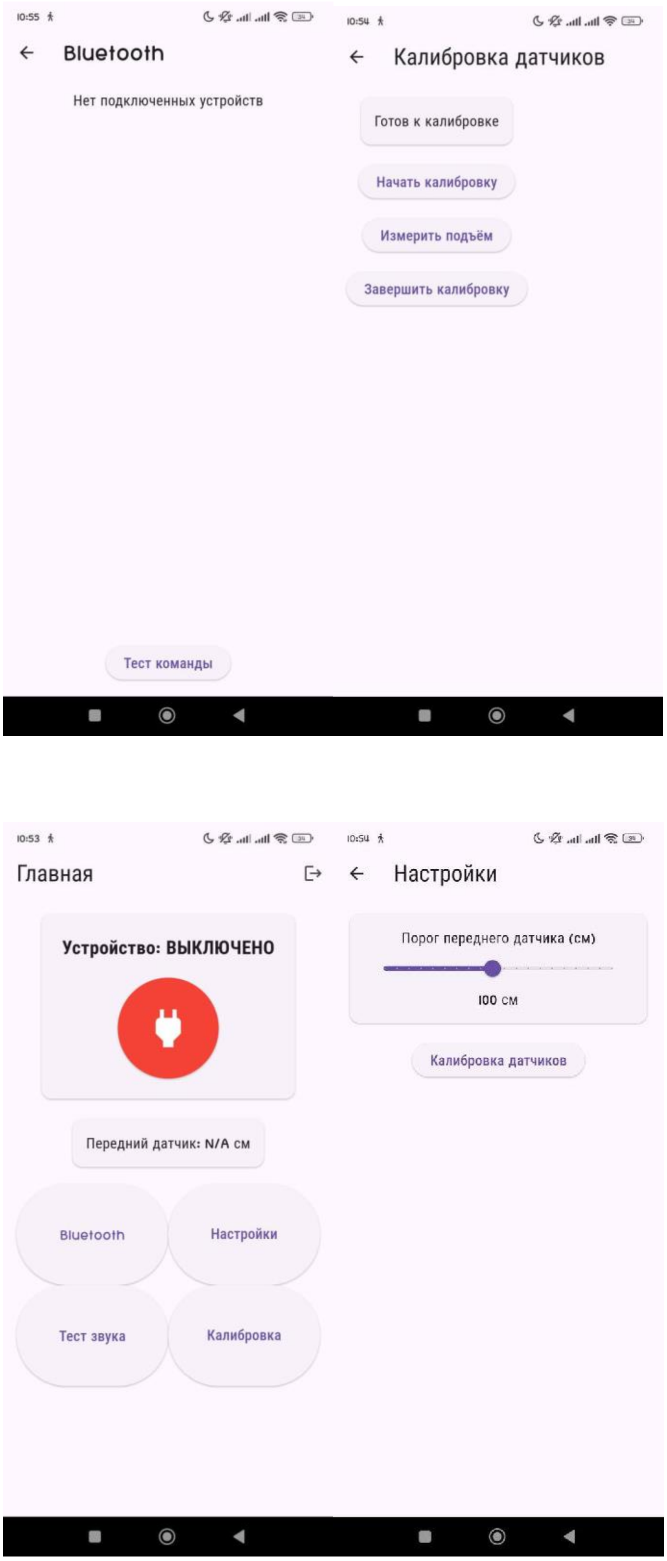


```
#include <SoftwareSerial.h>
#define TRIG_PIN 9
#define ECHO_PIN 10
#define BUZZER_PIN 8
SoftwareSerial BTSerial(2,3)
long duration;
int distance;
int threshold=50;
bool systemOn=true;

void setup(){
  pinMode(TRIG_PIN,OUTPUT);
  pinMode(ECHO_PIN,INPUT);
  pinMode(BUZZER_PIN,OUTPUT);
  BTSerial.begin(9600);
  Serial.begin(9600);
}

void loop(){
  if(BTSerial.available()){
    String cmd=BTSerial.readStringUntil('\n');
    cmd.trim();
    if(cmd.startsWith("THRESHOLD")){
      int val=cmd.substring(9).toInt();
      if(val>0&&val<=400){threshold=val;BTSerial.println("Threshold set to
"+String(threshold)+" cm");}
    }else if(cmd=="ON"){systemOn=true;BTSerial.println("System ON");}
    else if(cmd=="OFF"){systemOn=false;BTSerial.println("System OFF");}
  }

  if(systemOn){
    digitalWrite(TRIG_PIN,LOW);
    delayMicroseconds(2);
    digitalWrite(TRIG_PIN,HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG_PIN,LOW);
    duration=pulseIn(ECHO_PIN,HIGH);
    distance=duration*0.034/2;
    if(distance>0&&distance<=threshold){
      int delayTime=map(distance,0,threshold,50,500);
      digitalWrite(BUZZER_PIN,HIGH);
      delay(delayTime);
      digitalWrite(BUZZER_PIN,LOW);
      delay(delayTime);
    }else digitalWrite(BUZZER_PIN,LOW);
  }else digitalWrite(BUZZER_PIN,LOW);
  delay(50);
}
```

Figure 24

13 Appendix B

Mobile application interface



Нет подключенных устройств

Устройство: ВЫКЛЮЧЕНО

Передний датчик: N/A см

Порог переднего датчика (см)

100 см

Figure 25

App full source code: https://github.com/Patricl123/shoes-for-blind.git

Youtube video QR

URL: https://youtu.be/6K60dasWJ3I?feature=shared

Github Repository QR

URL: https://github.com/reem0110/A000_Obstacle_Detection_System_For_Visually_Impaired_People